Bilevel Network Design

Martine Labbé, Patrice Marcotte

Bilevel Network Design

Martine Labbé and Patrice Marcotte

Abstract This chapter is devoted to network design problems involving conflicting agents, referred to as the designer and the users, respectively. Such problems are best cast into the framework of bilevel programming, where the designer anticipates the reaction or rational users to its course of action, and fits many situations of interest. In this chapter, we consider four applications of very different nature, with a special focus on algorithmic issues.

1 Introduction

The framework of bilevel programming allows the modelling of hierarchical situations where a leader anticipates the rational reaction of a non-cooperating follower whose objective and/or constraints are influenced by the leader's decisions. In the context of network design, this paradigm is especially relevant when the designer of a network does not have a direct control of user flows, who are assigned according to their own logic. In this chapter, we illustrate, through four distinct applications, the modelling and algorithmic issues that characterize bilevel network design problems. Throughout, we assign a broad sense to the term 'network design', meaning any program that involves the determination of variables that impact the structure of a graph or a network.

Martine Labbé
GOM, Université Libre de Bruxelles, Belgium, and INRIA, Lille, France, e-mail: mlabbe@ulb.ac.be

Patrice Marcotte
CIRRELT and DIRO, Université de Montréal, Canada, e-mail: marcotte@iro.umontreal.ca

2 A primer on bilevel programming

Bilevel programs, of which the well-known Stackelberg games are a particular instance, involve a leader and a follower acting in a hierarchical fashion. In our framework, the leader makes decisions embodied into a vector $x^1 \in R^{n_1}$, anticipating the reaction $x^2 \in R^{n_2}$ of the follower to its design x^1. The general formulation of a bilevel program is as follows:

$$\begin{array}{ll} \max\limits_{x^1,x^2} & f_1(x^1,x^2) \\ \text{subject to} & (x^1,x^2) \in X_1 \\ & x^2 \in S_2(x^1)\,, \end{array} \tag{1}$$

where, for a given design vector x^1, $S_2(x^1)$ is the set of optimal solutions of the follower's problem, i.e.,

$$S_2(x^1) = \arg \min_{y^2 \in X_2(x^1)} f_2(x^1,y^2)\,. \tag{2}$$

Under standard assumptions such as compactness or continuity, one can prove the existence of at least one solution to a bilevel program. What makes the problem difficult is that, whenever the lower level problem is not trivial, no closed form solution is available for x_2 as a function of x_1.

Whenever the set $S_2(x^1)$ contains more than one element, the above formulation implies that the leader is free to select the element that maximizes its objective. This is the 'optimistic' case, in contrast with the 'pessimistic' case where the leader assumes that the follower will select the element of $S_2(x^1)$ that minimizes the leader's objective. Throughout this chapter, we assume that the optimistic case prevails.

Now, for notational convenience, we set

$$X_2(x^1) = \{x^2 : (x^1,x^2) \in X_2\}$$

for some set $X_2 \subseteq R^{n_2}$ and record a bilevel program in the following 'vertical' format, which contains all relevant information:

$$\begin{array}{ll} \max\limits_{x^1} & f_1(x^1,x^2) \\ \text{subject to} & (x^1,x^2) \in X_1 \\ & \\ \min\limits_{x^2} & f_2(x^1,x^2) \\ \text{subject to} & (x^1,x^2) \in X_2\,. \end{array} \tag{3}$$

It is important to understand that the follower is oblivious to the 'upper level' constraint $(x^1,x^2) \in X_2$. Its enforcement is the sole responsibility of the leader, which

must make sure that it is satisfied by the rational reaction of the follower. We now turn our attentions to generic algorithmic frameworks. These are mostly based on reformulations as single-level programs, which can be achieved in a variety of ways. We introduce the three most common ones.

Value function formulation

In this formulation, lower level optimality is enforced by specifying that, for a given design vector x^1, x^2 must be lower-level feasible, and outperform any alternative solution. This yields the nonconvex semi-infinite program

$$\begin{aligned} \max_{x^1,x^2} \quad & f_1(x^1,x^2) \\ \text{subject to} \quad & (x^1,x^2) \in X_1 \cap X_2 \\ & f_2(x^1,x^2) \leq f_2(x^1,y^2) \quad \forall y_2 \in X_2(x^1)\ . \end{aligned} \tag{4}$$

First-order conditions formulation

If the function f_2 is differentiable and convex with respect to x^2, and if the sets $X_2(x_1)$ are convex for every feasible x_1, one can characterize the set $S_2(x^1)$ via the first-order necessary and sufficient optimality conditions of the lower level programs. This yields the single-level formulation

$$\begin{aligned} \max_{x^1,x^2} \quad & f_1(x^1,x^2) \\ \text{subject to} \quad & (x^1,x^2) \in X_1 \cap X_2 \\ & \langle \nabla_2 f_2(x^1,x^2), x^2 - y^2 \rangle \leq 0 \quad \forall y^2 \in X_2(x^1)\ , \end{aligned} \tag{5}$$

where ∇_2 represents the gradient of f_2 with respect to its second argument x^2.

Kuhn-Tucker-based formulation

If the set X_2 is expressed by a set of inequalities

$$X_2 = \{(x^1,x^2) : g_{2i}(x^1,x^2) \leq 0,\ i = 1,\ldots,m_2\},$$

for some convex functions g_{2i}, the follower's objective f_2 is convex with respect to x^1, *and* a regularity condition (constraint qualification) holds, then $S_2(x^1)$ can be replaced by the necessary and sufficient Kuhn-Tucker optimality conditions of the lower level program. This yields the single-level formulation

BILKT :

$$\begin{array}{rll} \max\limits_{x^1,x^2} & f_1(x^1,x^2) & \\ \text{subject to} & (x^1,x^2) \in X_1 \cap X_2 & \\ & \nabla_2 f_2(x^1,x^2) + \sum\limits_{i=1}^{m_2} \lambda_i \nabla g_{2i}(x^1,x^2) = 0 & \\ & \lambda_i g_{2i}(x^1,x^2) = 0 & i = 1,\ldots,m_2 \\ & \lambda_i \geq 0 & i = 1,\ldots,m_2\ , \end{array} \tag{6}$$

that, in contrast with the preceding two formulations, involves but a finite number of constraints. Note that BILKT, which might be the most frequently used in practice, exhibits the combinatorial nature of the problem through the complementarity between the vector of Lagrange multipliers y and the inequality constraints defined by the convex functions g_{2i}'s. A convenient shortcut for the complementarity system formed by the last two constraints of BILKT is

$$\lambda \geq 0 \quad \perp \quad g_2(x^1,x^2) \leq 0,$$

which stresses the relationship between bilevel programming and the class of problems known as Mathematical Programs with Equilibrium Constraints, or MPEC's.

Common to all three formulations is that their constraints do not satisfy any constraint qualification generically, hence standard techniques of nonlinear (nonconvex) programming may fail to produce even stationary points. It follows that efficient algorithms must explicitly deal with the twin continuous-combinatorial nature of the problem, and frequently adapt to the specific nature of the application under consideration.

Equilibrium constraints

In several applications of interest, a population of players reacts to the leader's decision by achieving an equilibrium. This occurs for instance in n-player games, where a Nash equilibrium is reached when no player can increase its objective by acting alone. Another important application involves selfish users that strive for minimum travel time in a congested network. A Wardrop equilibrium of this non-atomic game corresponds to a multicommodity flow vector x such that, for every origin-destination pair, flow is only assigned to paths that are shortest with respect to the delays induced by x.

In general, an equilibrium is not naturally the solution of an optimization problem. Rather, the set of lower-level equilibria corresonding to an upper-level vector x^1 can be characterized as the solution of a variational inequality involving a mapping $F_2 : R^{n_2} \to R^{n_2}$, i.e.,

$$S_2(x^1) = \{x^2 \in X_2(x^1) : \langle F_2(x^1,x^2), x^2 - y^2 \rangle \leq 0 \quad \forall y^2 \in X_2(x^1)\}.$$

If F_2 happens to be the gradient of some function f_2, then the solutions of the above variational inequality are simply the vectors that satisfy the first-order optimality

conditions of a lower level convex program. The added generality occurs when this is not the case, i.e., for a continuously differentiable mapping F_2, when its Jacobian matrix fails to be symmetric. The equivalent of the Kuhn-Tucker formulation is then obtained by substituting F_2 to ∇_2 in BILKT.

Solution algorithms

In their generality, bilevel programs are akin to unstructured global optimization problems, and their feasible space may even be disconnected in the presence of upper level constraints involving the vector x^2. Actually, their strong NP-hardness has been proven in the 'simple' case where all functions and constraints are linear.

Most algorithms rely on a single-level reformulation and, for tractability, assume that the lower level problem is relatively easy to solve. Initiated with a relaxed problem where the leader controls both x^1 and x^2, formulations (4) and (5) suggest a cutting-plane approach that only generates 'optimality' or 'variational' constraints as required. At each iteration, and for a given upper level vector x^1, the most violated constraint is appended to the relaxed problem. For the value function formulation, this amounts to solving the lower level problem

$$\min_{y^2 \in X_2(x^1)} f_2(x^1, y^2)$$

while, for the 'first-order' formulation, one solves

$$\min_{y^2 \in X_2(x^1)} \langle \nabla_2 f_2(x^1, x^2), y^2 \rangle,$$

which reduces to a standard linear program if the lower level constraints are linear. Note that, in both cases, the relaxed problems are highly nonlinear and nonconvex.

Formulation BILKT might be the most interesting from an algorithmic point of view, the main issue being the treatment of its complementarity constraints. Indeed, their presence makes the problem 'irregular', i.e., constraint qualifications that are the basis of convergence analysis are not generically satisfied. This difficulty can be sidestepped by either appending a multiple of the complementarity term into the objective[1] or by smoothing the complementarity term, for instance

$$\lambda_i g_{2i}(x^1, x^2) \leq \varepsilon$$

for some small scalar ε. A third approach, which highlights the combinatorial nature of the complementarity constraints, consists in replacing the complementarity constraint $\lambda_i g_{2i}(x^1, x^2) = 0$ by the more tractable triple

$$\begin{aligned} \lambda_i &\leq M u_i \\ g_{2i}(x^1, x^2) &\leq M(1 - u_i) \\ u_i &\in \{0, 1\}, \end{aligned}$$

[1] Under weak assumptions, the penalty is 'exact', i.e., a global solution of the penalized problem involving a large but finite multiplier is a global solution of the original bilevel program. Unfortunately, this result might fail to hold for local solutions.

for some 'large' constant M. When the constraints are linear and the objectives are linear (respectively quadratic), the resulting mixed integer program can be solved to global optimality by a dedicated software. Also noteworthy is that, when the vector of binary variables is fixed, the resulting program possesses a structure (a network structure for instance) that can be exploited.

Another approach, fruitful in nonlinear programming, is to approximate the original program by a related model. Inspired by the trust region framework, one can rely on an approximation involving linear or quadratic functions, which can be solved for a global optimum. Alternatively, one can design an approximation that exploits the problem's structure, as we will see in the next sections.

Finally, various heuristics can be applied. These can be either of a generic nature, or based on the key features of the problem under investigation. On the generic front, meta-heuristics that have been proposed for combinatorial problems (simulated annealing, tabu search, genetic algorithms, etc.) can exploit the presence of the binary variables u_i's, as well as the relative ease with which the program associated with fixed u-values can be solved.

3 The continuous network design problem

Consider the problem that consists in improving a subset of links of an urban transportation network, with the aim of minimizing the weighted sum of transportation and investment costs. Since users minimize their individual cost, their objective is not aligned with that of the network builder. More precisely, for a given link improvement vector $z \in Z$, the arc flow vector x achieves Wardrop equilibrium that satisfies the variational inequality

$$\langle F(x,z), x-y\rangle \leq 0 \quad \forall y \in X,$$

where x is the link-flow vector, F the associated cost (delay) mapping, and X represents the set of feasible arc flows. In this realm, the designer of the network faces the bilevel problem (an MPEC to be precise)

$$\begin{array}{rl} \min\limits_{z,x} & \langle F(x,z),x\rangle + c(z) \\ \text{subject to} & z \in Z \\ & x \in X \\ & \langle F(x,z), x-y\rangle \leq 0 \quad \forall y \in X \quad , \end{array} \tag{7}$$

where $c(z)$ denotes the cost of implementing design z.

We now focus on the case where z is only constrained to be nonnegative, and where the delay and investment functions are link-separable, thus are gradient mappings. Letting $\mathscr{A}$ denote the index set of arcs, this yields the bilevel program

$$\begin{array}{ll} \min\limits_{z} & \sum\limits_{a\in\mathscr{A}} [F_a(x_a,z_a)x_a + c_a(z_a)] \\ \text{subject to} & z_a \geq 0 \qquad a \in \mathscr{A} \\ \\ \min\limits_{x} & \sum\limits_{a\in\mathscr{A}} \int_0^{x_a} F_a(t,z_a)\,dt \\ \text{subject to} & x \in X\,. \end{array} \tag{8}$$

Furthermore, we assume that the link delay functions F_a are of the form

$$F_a(x_a,z_a) = F_a(x_a/z_a),$$

for some nonnegative, convex and increasing functions F_a.

From the theoretical point of view, the problem is NP-hard. However, in the view that the leader and follower's objectives are not at odds[2], it is tempting to let the leader control both the design and flow variables, yielding a design vector z and the corresponding equilibrium flow $x(z)$. More specifically, let us consider the system-optimal problem

$$\min_{z\geq 0, x\in X} \sum_{a\in\mathscr{A}} [F_a(x_a/z_a)x_a + c_a(z_a)]. \tag{9}$$

For a given flow vector x, the problem becomes separable in z, with $z_a(x_a)$ being the unique solution of the parameterized equation

$$-(x_a/z_a)^2 F_a'(x_a/z_a) + c_a'(z_a) = 0, \tag{10}$$

obtained by setting to zero the gradient of the leader's objective. One then solves the mathematical program

$$\min_{x\in X} \sum_{a\in\mathscr{A}} [F_a(x_a/z_a(x_a))x_a + c_a(z_a(x_a))], \tag{11}$$

to obtain feasible design vector $\bar{z}$, together with a flow vector that might not be in equilibrium. A feasible solution can then easily be recovered by setting the flow vector to the equilibrium corresponding to the $\bar{z}$, say $\bar{x}$. This simple heuristic procedure will be denoted H1.

We now further restrict our attention to polynomial delay and investment functions, i.e.,

$$F_a(x_a/z_a) = \alpha_a + \beta_a (x_a/z_a)^p$$

and

$$c_a(z_a) = l_a z_a^m,$$

for some scalars $p \geq 1$, $m \geq 0$ and $l_a \geq 0$. According to these assumptions, the root of Equation (10) can be obtained in closed form. Furthermore, if the cost func-

[2] The leader's objective involves total transportation costs, versus marginal costs for the follower.

tion c_a are convex (respectively concave), then problem (11) is convex (respectively concave). The concave situation occurs when $m < 1$, and yields an extremal flow solution. In the particular case where functions c_a's are linear, (11) reduces to a linear multicommodity flow problem that can be easily solved by shortest path computations.

Another heuristic approach to the problem, denoted H2, consists in iteratively solving Equation (11) for fixed z-vector, and then performing an equilibrium assignment, for fixed design vector z. Whenever this cobweb process converges, it will do so at a couple $(\bar{x},\bar{z})$ where Equation (10) is satisfied, while $\bar{x}$ is in equilibrium with respect to z.

A third heuristic strategy (H3) consists in adjusting the capacity vector to the system-optimal flows $\bar{x}$. Algorithms H2 and H3 are actually subsumed by a more general scheme H4 where one solves the single-level program

$$\min_{z\geq 0, x\in X} \sum_{a\in A} \left[\int_0^{x_a} F_a(t,z_a)\,dt + \xi_a c_a(z_a) \right] \tag{12}$$

for some set of nonnegative weights ξ_a's. It is clear that any solution to that program yields equilibrium flows with respect to z. Moreover, it can be shown that there exists a set of weights ξ_a such that an optimal solution of (12) is also optimal for the original bilevel program. While finding such weights is theoretically as hard as solving the continuous network design problem in the first place, educated guesses, such as setting ξ to the vector of ones may prove of interest. Actually, in the view that (12) can be solved quickly, one might solve (12) over a range of weight vectors with identical values $\bar{\xi}$ for instance. Setting all parameters ξ_a's to $1/(p+1)$ yields H2, while setting them to $p+1$ yields H3.

An interesting feature of the model is that information about worst-case bounds of the heuristics can be derived with respect to key parameters of the problem, namely the respective degrees p and m of the delay and cost polynomials. Let us denote, for a given heuristic H (H1, H2, H3 or H4), by $\rho^{\mathrm{H}}(m,p)$ the worst-case ratio of the cost of the heuristic solution value over that of the unknown optimal value, i.e.

$$\rho^{\mathrm{H}}(m,p) = \sup \frac{\text{cost of heuristic solution}}{\text{cost of optimal solution}}, \tag{13}$$

where the supremum is taken over all possible values of the remaining parameters and network configurations. We have the following results:

- $\lim_{p\to\infty} \rho^{\mathrm{H1}}(p,1) \geq 2$
- $\rho^{\mathrm{H2}}(p,1) = p+1$
- $\rho^{\mathrm{H3}}(p,m) = \dfrac{m(p+1)}{m+p} + \dfrac{p}{(m+p)(p+1)^{m/p}}$
- $1 + \dfrac{p}{\xi(p+1)} \leq \rho^{\mathrm{H4}}(p,m) \leq \dfrac{\xi^{p/p+1}}{(p+1)^{1/p+1}} \left[1 + \dfrac{p}{\xi(p+1)}\right]^2$.

It is worth noting that, for $p = m = 1$, the worst-case bound of heuristic H3 is equal to 5/4, which is less than the price of anarchy[3] for affine latency functions. This bound can also be improved to $49/41 \approx 1.195$ by computing the best solution provided by H3 and a closely related method, a result that was extended to a larger class of delay functions than polynomials. Worst-case results indicate that Heuristic H3 outperforms both H1 and H2. However, computational results tend to show that H1 and H2 perform much better in practice.

A straightforward extension is concerned with the improvement of existing networks. Let us denote by z^0 the vector of initial link capacities. This more general model is subsumed by the 'standard' model if one sets the investment cost to the nondifferentiable function

$$c_a \times \max\{0, z_a - z_a^0\}.$$

While the latter can be approximated as closely as desired by a differentiable function, allowing the implementation of previously described heuristics, the worst-case results do not hold any more.

4 A competitive location-queuing model

Besides its intrinsic interest, this topic provides an opportunity to analyze algorithmic techniques for addressing complex mathematical programs involving both discrete and continuous variables. In the model considered, a firm makes decisions concerning the location and service levels of facilities, with the aim of attracting the maximum number of customers. At the lower level, customers minimize their individual disutility.

Before providing a mathematical description of the model, we broadly describe the supply-demand setting.

Facilities fall into two categories, namely those owned by the leader and its competitors, respectively. Each open facility is characterized by its location vertex and service level, which are the decision variables of the leader firm. Service is exponentially distributed, and queue length limited to a predetermined capacity. Whenever the capacity is exceeded, balking occurs, i.e., customers are denied service. A consequence of this phenomenon is that the model makes sense even if arrival rates exceed service rates. Throughout, we assume that the location and service rates of the competition are fixed and known.

The demand side is characterized by Poisson processes associated with nodes of the network. For given facility locations and service levels, users are assigned to the facilities according to a discrete choice (logit) model where random utilities are linear combinations of travel time to facilities, queueing delays, probability of *not* accessing service, and a random term that makes for unobserved features. In

[3] The price of anarchy is defined as the worst-case ratio of the true delay associated with an equilibrium over system-optimal delay.

this framework, the objective of the leader is to maximize the number of customers served at its facilities, subject to a budget constraint.

Let us now focus on the assignment part. We denote by d_i the demand rate originating from node i, by λ_j the arrival rate at facility j, and by K_j the capacity of facility j. We assume that users are rational and minimize a disutility expressed as a positive linear combination

$$u_{ij} = t_{ij} + \alpha w_j + \beta p_{K_j} + \varepsilon,$$

of travel time t_{ij}, waiting time w_j, probability p_{K_j} of not accessing facility j, plus a random Gumbel term ε with cumulative distribution function $\exp(-\exp(x/\theta))$. This yields a closed form expression for the assignment of users to facilities:

$$x_{ij} = d_i \frac{e^{-\theta u_{ij}}}{\sum\limits_{l \in J^*} e^{-\theta u_{il}}} \quad , \tag{14}$$

where J^* represents the set of open facilities. Classical queueing theory results yield the expressions

$$w_j = \frac{1}{\mu_j} \left(K_j + \frac{K_j}{(\lambda_j/\mu_j)^{K_j} - 1} - \frac{1}{(\lambda_j/\mu_j) - 1} \right)$$

and

$$p_{K_j} = \frac{\lambda_j}{\mu_j} \cdot \frac{1 - (\lambda_j/\mu_j)}{1 - (\lambda_j/\mu_j)^{K_j+1}} \quad .$$

If the process intensity λ_j/μ_j is equal to 1, the above expressions are replaced by their limits, which are finite.

Let J_1 denote the index set of the leader's facilities. The aim of the leader is to maximize the number of customers that access its facilities, i.e.,

$$\sum_{j \in J_1} \lambda_j (1 - p_{K_j}),$$

through the control of the service rates μ_j ($j \in J_1$) and location decisions, represented by binary variables y_j set to 1 if a facility is located at node j of the network, and to 0 otherwise. Denoting by c_j^f the fixed cost of opening a facility at node j, by c_j the cost of providing a unit of service level at node j, and by B the total budget constraint, the problem can be formulated as the mathematical program

$$
\begin{aligned}
\text{LEADER:} \quad & \max_{y,\mu} && \sum_{j\in J_1} \lambda_j (1-p_{K_j}) && \\
& \text{subject to} && \sum_{j\in J_1} c_j^f y_j + \sum_{j\in J_1} c_j \mu_j \le B && \\
& && \mu_j \le M y_j && j \in J_1 \\
& && y_j \in \{0,1\} && j \in J_1 \\
\text{USERS:} \quad & && x_{ij} = d_i y_j \frac{e^{-\theta\left(t_{ij}+\alpha w_j+\beta p_{K_j}\right)}}{\sum_{l\in J^*} e^{-\theta\left(t_{il}+\alpha w_l+\beta p_{K_l}\right)}} && i \in I, j \in J \\
& && \lambda_j = \sum_{i\in I} x_{ij} && j \in J \\
& && w_j = \frac{1}{\mu_j}\left(K_j + \frac{K_j}{(\lambda_j/\mu_j)^{K_j}-1} - \frac{1}{(\lambda_j/\mu_j)-1}\right) && j \in J \\
& && p_{K_j} = (\lambda_j/\mu_j)^{K_j} \frac{1-(\lambda_j/\mu_j)}{1-(\lambda_j/\mu_j)^{K_j+1}} && j \in J \quad .
\end{aligned}
$$

where the sole decision variables y and μ have been emphasized under the 'max' operator, and M is a suitably large 'big-M' constant. In the limiting case $\theta = \infty$, the first user equation reduces to the complementarity system (Wardrop principle)

$$
t_{ij} + \alpha w_j(x,\mu) + \beta p_{Kj}(x,\mu) \begin{cases} = \xi_i, \text{ if } x_{ij} > 0 \\ \\ \ge \xi_i, \text{ if } x_{ij} = 0 \quad . \end{cases} \tag{15}
$$

While the above looks like a standard optimization program, its main difficulty resides in the highly nonlinear constraints that define the user flows. Indeed, since the quantities w_j and p_{K_j} both depend on x_{ij}, it follows that the first user constraint is actually a fixed point equation that is not trivial to solve in its own right. To cast the problem into a more convenient framework, we first note that any solution of the optimization program (convex in the user variables)

$$
\begin{aligned}
\min_{x} \quad & \sum_{i\in I}\sum_{j\in J^*}\left(\frac{1}{\theta}x_{ij}\ln x_{ij} + t_{ij}x_{ij}\right) + \alpha \sum_{j\in J^*}\int_0^{\lambda_j} w_j(\lambda,\mu_j)d\lambda \\
& \qquad\qquad + \beta \sum_{j\in J^*}\int_0^{\lambda_j} p_{Kj}(\lambda,\mu_j)d\lambda \qquad\qquad (16) \\
\text{subject to} \quad & \sum_{j\in J^*} x_{ij} = d_i && i \in I \\
& x_{ij} \ge 0 && i \in I, j \in J^* \\
& \lambda_j = \sum_{i\in I} x_{ij} && j \in J^*
\end{aligned}
$$

yields an equilibrium solution. When $\theta = \infty$, the first term simply disappears, and users are assigned to shortest paths.

Upon replacing the fixed point equation by the convex program, one obtains a bilevel program involving a structure that can be exploited algorithmically. For instance, the first term of the lower level objective is convex, the second is jointly pseudo-convex in λ and μ. Furthermore, in the important case when no balking occurs ($K_j = \infty$ for all leader facilities), the third term is jointly convex in λ and μ.

Let us first consider the no-balking situation. In this case, $p_{K_j} = 0$ and, in order that the problem be meaningful, the total service rate must exceed total demand. This can be already satisfied by the competing facilities, or enforced through additional bound constraints on service levels. This allows the following strategy for solving the bilevel location problem.

- Approximate the lower level objective by a (convex) piecewise linear function.
- Express the approximate lower level program as a linear program.
- Replace the linear program by its primal-dual optimality conditions. This yields linear constraints, with the exception of primal-dual complementarity.
- With the help of binary variables, linearize the complementarity constraints to obtain a mixed integer program (MIP).
- Let y^* and μ^* be the partial solution of the MIP.
- Compute the equilibrium flows x^* compatible with y^* and μ^*.

One can show that, as the mesh of the piecewise linear approximation decreases, the solution of the MIP converges to a solution of the original problem. More important, it was observed that a coarse mesh was actually sufficient to yield high quality solutions. If the parameters K_j are finite, the situation becomes more complex, and one has to introduce additional binary variables to linearize both the upper and lower level objectives. Nevertheless, the same algorithmic strategy can still be applied.

Since the curse of dimensionality hits very early for such complex bilevel programs, it warrants looking for alternative algorithmic approaches. One such approach consists, as has been proposed for the continuous version of the network design problem considered in the previous section, to replace the bilevel program by a single level proxy. In this regard, an equivalent to H1 would be to let the leader control all decision variables. However, this yields a poor approximation, whose trivial solution is to first build a single facility (the one with least fixed cost), to spend the residual budget on its service level, and to direct the maximal amount of customers to this facility, without regards for the competition. A better alternative is, similar to H2, to minimize a proxy program that yields equilibrium flow patterns, for instance

$$
\begin{aligned}
\min_{y,\mu,x} \quad & \sum_{i\in I}\sum_{j\in J^*}\left(\frac{1}{\theta}x_{ij}\ln x_{ij}+t_{ij}x_{ij}\right)+\alpha\sum_{j\in J^*}\int_0^{\lambda_j} w_j(\lambda,\mu_j)d\lambda \\
& \qquad\qquad +\beta\sum_{j\in J^*}\int_0^{\lambda_j} p_{Kj}(\lambda,\mu_j)d\lambda \\
\text{s.t.} \quad & \sum_{j\in J} x_{ij} = d_i && i\in I \\
& \sum_{j\in J_1} c_j^f y_j + \sum_{j\in J_1} c_j\mu_j \le B \\
& \lambda_j = \sum_{i\in I} x_{ij} && j\in J \\
& y_j\in\{0,1\} && j\in J \\
& x_{ij}\ge 0 && i\in I, j\in J \quad .
\end{aligned}
\tag{17}
$$

In the spirit of H4 introduced in the previous section, the objective can be generalized by adding a term that depends on the service level vector μ, yielding:

$$
\begin{aligned}
\min_{y,\mu,x} \quad & \sum_{i\in I}\sum_{j\in J^*}\left(\frac{1}{\theta}x_{ij}\ln(x_{ij})+t_{ij}x_{ij}\right)+\alpha\sum_{j\in J^*}\int_0^{\lambda_j} w_j(\lambda,\mu_j)d\lambda \\
& \qquad\qquad +\beta\sum_{j\in J^*}\int_0^{\lambda_j} p_{Kj}(\lambda,\mu_j)d\lambda+\sum_{j\in J_1}\xi_j\mu_j \quad ,
\end{aligned}
\tag{18}
$$

subject to the same constraints. Note that, in the important situation where $K=\infty$ (no balking), this yields a simple convex program. Otherwise, a nonconvex piecewise linear approximation of its objective results in a mixed integer formulation that can be solved to global optimality.

While it can be proved that there exists an assignment of the ξ_j variables that yields an optimal solution of the original problem, finding such assignment is theoretically as difficult as solving the original problem. Nevertheless, based on the intuition that it makes sense for the leader to favor facilities located close to high demand nodes, the educated guess

$$\xi_j = d_i/t_{ij}$$

has been proposed. If demand nodes coincide with facility location, a small number can be added to the denominator, to avoid dividing by zero. The formula can also be enhanced to take into account the fixed costs c_j^f, for instance,

$$\xi_j = -d_i/(c_j^f t_{ij}) \quad .$$

5 Network pricing

Pricing offers a rich area for applications of bilevel programming. This section discusses such problems involving an underlying network structure. More precisely, we consider the problem that consists in setting tolls on a subset of arcs of a multicommodity transportation network, with the aim of maximizing profit. In this context, the toll manager anticipates the flows that result from its toll policy, i.e., users are assigned to paths that minimize their respective disutility, taking into account constant transportation costs, as well as out-of-pocket costs.

As an example, consider the network depicted in Figure 1, that involves two toll arcs (dotted) and a single user that travels from node 1 to node 5. The cost (equal to 22) of the toll free path from node 1 to node 5 provides an upper bound on the maximum amount the user agrees to pay for its trip. Further the transportation cost is at least 6, which corresponds to the cost of a shortest path when both tolls are set to zero. This yields an upper bound of $22-6=16$ on the revenue. Interestingly, this bound cannot be reached since the optimal solution is obtained by setting the toll on arc $(2,3)$ to 5 and the toll on arc $(4,5)$ to 10.

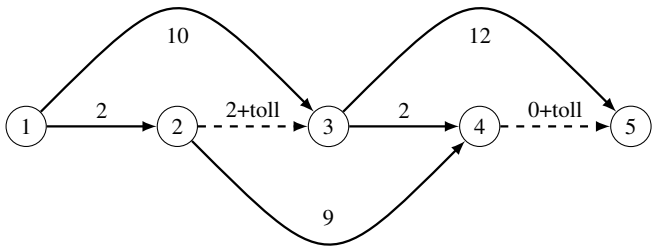

Fig. 1 A toll pricing network and its arc costs.

In general, the multicommodity transportation network is characterized by a graph $\mathscr{G}$ with node set $\mathscr{N}$ and arc set $\mathscr{A}$. Each arc $a \in \mathscr{A}$ is endowed with a cost c_a. The set of arcs $\mathscr{A}$ is partitioned into two subsets $\mathscr{A}_1$ and $\mathscr{A}_2$, the former containing the arcs controlled by the toll manager and the latter the so-called 'toll free' arcs. The commodities $k \in \mathscr{K}$ represents groups of users willing to travel from the same origin $o^k \in \mathscr{N}$ to the same destination $d^k \in \mathscr{N}$. For commodity k, its demand is denoted by η^k and its nonnegative transportation cost on arc a by c_a^k.

The Network Pricing Problem (NPP) can be formulated as the following bilevel program that involves bilinear objectives and linear constraints :

NPP:

$$\begin{array}{lll} \max\limits_{t} & \sum\limits_{k\in\mathscr{K}} \eta^k \sum\limits_{a\in\mathscr{A}_1} t_a x_a^k & \\ \text{subject to} & t \in T & \\ \min\limits_{x} & \sum\limits_{k\in\mathscr{K}} (\sum\limits_{a\in\mathscr{A}_1} (c_a^k + t_a) x_a^k + \sum\limits_{a\in\mathscr{A}_2} c_a^k x_a^k) & \\ \text{subject to} & x^k \in X^k & k \in K, \end{array} \tag{19}$$

where X^k denotes the polyhedron of feasible flows for commodity k, and T imposes constraints, such as bounds, on the set of feasible tolls. In order that the profit be bounded, there must exist at least one toll free path, i.e., containing only arcs belonging to $\mathscr{A}_2$, for each origin-destination pair (commodity). In other words, the removal of toll arcs must leave all origin-destination pairs connected.

In NPP, we assume that the set T consists in all non negative toll values[4]. Further, X^k represents the set of feasible paths associated with commodity k, which is characterized by the following flow conservation constraints on variables x_a^k:

$$\sum_{a\in\delta(i)^+} x_a^k - \sum_{a\in\delta(i)^-} x_a^k = b_i^k \qquad i \in \mathscr{N} \tag{20}$$

$$x_a^k \geq 0 \qquad a \in \mathscr{A}, \tag{21}$$

where $\delta(i)^+$ (resp. $\delta(i)^-$) denotes the set of arcs having node i (resp. j) as tail (resp. head).

From the complexity point of view, it has been shown that NPP is strongly NP-hard, and actually APX-hard, even in the single commodity case. Alternatively, consider an instance of NPP that involves but one toll arc (see Figure 2). A commodity k uses the toll arc only if its value is not larger than the difference, say M^k, between the value of the shortest path not using the toll arc and the value of the shortest path with the toll set to zero. It is easy to see that the optimal toll value will be equal to some M^k, which are in polynomial number. If commodities are ranked in decreasing values of the M^k, the corresponding revenue is obtained, for each such M^k, by multiplying M^k by $\sum_{k'\leq k} \eta^{k'}$. This yields a polynomial algorithm.

A closer look at the bilevel formulation of NPP shows that the lower level problem decomposes into shortest path problems, one for each commodity, that can be formulated as linear programs, since costs have been assumed nonnegative. To obtain a single level reformulation, one may replace the lower level problem of each commodity by its primal and dual constraints, together with a constraint imposing the equality of the primal and dual objective functions. This approach is the linear programming version of the Kuhn-Tucker-based formulation, see Section 2, and

[4] If negative tolls are allowed, optimal solutions might require setting tolls to negative values.

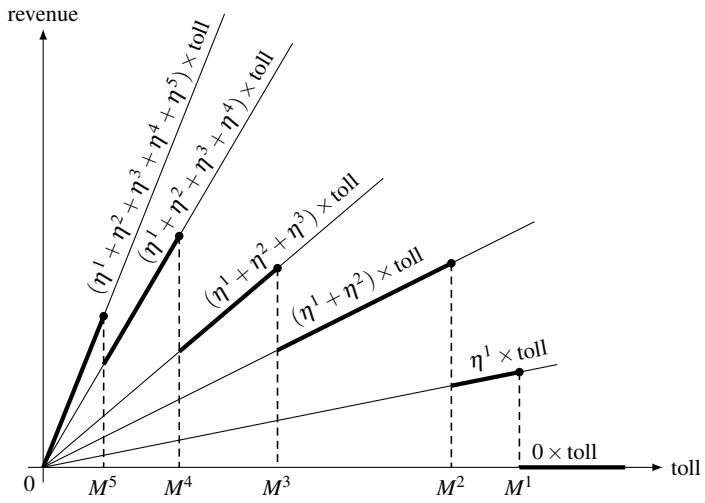

Fig. 2 Revenue function (in bold) for a single toll arc and 5 origin-destination pairs ($|\mathscr{K}| = 5$). The function is piecewise linear and continuous from the left. A local maximum is achieved at each 'critical' value M^k. In this instance, the optimum revenu is achieved when the toll is equal to M^4.

yields the following mixed integer linear formulation, in which the dual variables λ_i^k correspond to constraints (20).

MILP:

$$\begin{aligned}
\max_{t,x,\lambda} \quad & \sum_{k\in\mathscr{K}} \eta^k \sum_{a\in\mathscr{A}_1} t_a x_a^k & \\
\text{subject to} \quad & t_a \geq 0 & a\in\mathscr{A}_1\\
& \sum_{a\in\delta(i)^+} x_a^k - \sum_{a\in\delta(i)^-} x_a^k = b_i^k & i\in\mathscr{N}, k\in\mathscr{K}\\
& x_a^k \geq 0 & a\in\mathscr{A}\\
& \lambda_i^k - \lambda_j^k \leq c_a^k + t_a & a=(i,j)\in\mathscr{A}_1, k\in\mathscr{K}\\
& \lambda_i^k - \lambda_j^k \leq c_a^k & a=(i,j)\in\mathscr{A}_2, k\in\mathscr{K}\\
& \sum_{a\in\mathscr{A}_1} (c_a^k + t_a)x_a^k + \sum_{a\in\mathscr{A}_2} c_a^k x_a^k = \lambda_{o^k}^k - \lambda_{d^k}^k & k\in\mathscr{K}.
\end{aligned}$$

This model involves bilinear terms $t_a x_a^k$ in the objective and in some constraints. These terms can be linearized by using the standard technique consisting in substituting them by new variables, say p_a^k and appending the following new constraints, for all a in $\in \mathscr{A}_1$ and k in $\mathscr{K}$:

$$\begin{aligned} p^k &\leq M_a^k x_a^k \\ t_a - p_a^k &\leq N_a(1-x_a^k) \\ p_a^k &\leq t_a \\ t_a^k &\geq 0 \\ x_a^k &\in \{0,1\} \quad . \end{aligned} \tag{22}$$

Note that binary constraints regarding variables x_a^k are required in order to ensure the validity of the linearization. This is consistent with the fact that since these variables characterize shortest paths.

The MILP formulation can be improved by tightening the values of the big-M constants M_a^k and N_a, yielding an improved linear relaxation. Next, one may replace the original graph by an equivalent and usually smaller one, based on the observation that shortest paths are composed of alternating shortest subpaths containing only toll arcs or only toll free arcs. Finally, criteria have been proposed to eliminate some provably irrelevant variables x_a^k.

Despite these improvements, the size of instances for which solvers can address MILP is limited, thus prompting the development of (meta) heuristics, many of which iterate between path and toll phases. In this context, it is instructive to study the so-called inverse problem which consists in determining tolls maximizing the leader's revenue, under the assumption that the followers' paths are known. MILP then boils down to a linear program involving only toll variables. When there is only one follower (a single origin or a single destination), the inverse problem reduces to a shortest path problem on a modified graph.

The **Clique Pricing Problem** (CPP) is an NP-hard special case of NPP in which each commodity uses at most one toll arc. This framework fits the realm of a highway where the toll depends only on the entry and exit nodes used by the commodities, and re-entry is forbidden. The underlying network is then based on the complete graph on the highway entry and exit nodes whose arcs $a \in \mathscr{A}_1$ represent all subpaths of the original highway path (see Figure 3). The arcs of the OD clique represent shortest toll free paths between origin and destinations of commodities k and their cost is denoted by c_{od}^k. The coefficient c_a^k of an arc $a \in \mathscr{A}_1$ is the sum of three terms: the minimum cost from o^k to the tail node of a, the minimum cost from the head node of a to d^k (dashes arcs) and the cost for traversing arc a.

It follows that, for each commodity, the only relevant arc of $\mathscr{A}_2$ is (o^k, d^k). In consequence the k-th objective function of the lower level problem (19) then simplifies to

$$\min_{x \in X^k} \quad \sum_{a \in \mathscr{A}_1} (c_a^k + t_a) x_a^k + c_{od}^k x_{od}^k$$

and the set X^k to

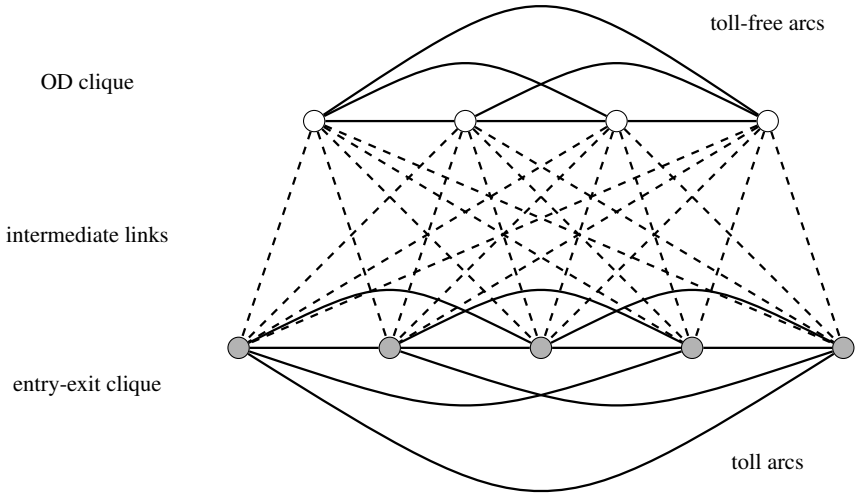

Fig. 3 Clique representation. All arcs are double-ended.

$$\sum_{a\in\mathscr{A}_1} x_a^k + x_{od}^k = 1 \quad k \in \mathscr{K}$$
$$x_a^k \geq 0 \quad a \in \mathscr{A}_1 \cup \mathscr{A}_2, k \in \mathscr{K}$$

This simple structure of X^k allows to reformulate CPP as a single level problem by stating explicitely that the objective function of the k-th commodity is less than or equal to the total cost corresponding to each arc a of the highway, i.e.,

$$\begin{aligned} \sum_{a\in\mathscr{A}_1} (c_a^k + t_a)x_a^k + c_{od}^k x_{od}^k \leq c_b^k + t_b \quad & b \in \mathscr{A}_1, k \in \mathscr{K} \\ \sum_{a\in\mathscr{A}_1} (c_a^k + t_a)x_a^k + c_{od}^k x_{od}^k \leq c_{od}^k \quad & k \in \mathscr{K} \quad , \end{aligned} \tag{23}$$

which corresponds to the value function formulation (4). A mixed binary linear formulation is obtained by substituting the bilinear terms $t_a x_a^k$ with variables p_a^k, and appending constraints (22) to the model.

For all $b \in \mathscr{A}_1$ and for all $\mathscr{S} \subset \mathscr{A}_1 \setminus \{b\}$, the inequality

$$\sum_{a\in\mathscr{A}_1} (c_a^k x_a^k + p_a^k) + c_{od}^k x_{od}^k \leq t_b + c_b^k + \sum_{a\in\mathscr{S}} (p_a^k + (c_a^k - c_b^k)x_a^k).$$

holds. Indeed, remembering that $x_a^k = 0$ implies that $p_a^k = 0$, the inequality reduces to (23) if $\sum_{a \in \mathscr{S}} x_a^k = 0$. If not, there must be exactly one arc, say b', such that $x_{b'}^k = 1$, and the inequality reads $c_{b'}^k + p_{b'}^k \leq t_b + p_{b'}^k + c_{b'}^k$. which is obviously satisfied.

The inclusion of these inequalities in MILP provides an ideal formulation in the (polynomial) case of a single commodity, i.e., its linear relaxation yields an optimal solution. In the general case involving several commodities, the inequalities strengthen significantly the formulation and can be separated in polynomial time.

Interestingly, CPP is equivalent to the classical product pricing problem in economics. In this problem the price of products $a \in \mathscr{A}_1$ must be determined in order to maximize revenue. Each customer $k \in \mathscr{K}$ is endowed with a reservation price R_a^k for each product a and buys the product a that maximizes $R_a^k - t_a$, provided that this 'utility' is non negative. The equivalence of the two problems is obtained by setting $R_a^k = c_{od}^k - c_a^k$.

When products constitute bundles or subsets of items it may be sensible to impose monotonicity and triangle inequality constraints. Monotonicity constraints specify that if the item set of product a is included in the item set product b, then $p_a \leq p_b$, and triangle inequality constraints stipulate that $p_c \leq p_a + p_b$ if the item set of c is the union of the item sets of a and b, which are disjoint. These type of constraints make particular sense for the clique pricing problem since a toll arc represents a subpath of the highway, i.e., a particular subset of its arcs. The inclusion of monotonicity and triangle inequalities into CPP results in a significant increase in the integrality gap, and consequently makes more challenging its numerical resolution.

We conclude this section by mentioning other variants of NPP. The first three address more complex models of user behaviour. The first variant integrates **elastic demand** at the lower level, which allows to dispense with the existence of toll free paths. In the **multiclass model**, the relative utility of delays and out-of-pocket costs varies across the population while, in the **discrete choice model**, a random term is added to the costs c_a and d_a. In those two cases, algorithms that rely on approximations that can be formulated as mixed integer programs have been proposed.

The fourth variant involves lower level **transshipment problem**. In the modified formulation of NPP, the commodity index k is dropped, and link capacities are introduced. Unfortunately, since it cannot be assumed that origin-destination flows are assigned to a unique path, the bilinear $t_a x_a$ cannot be linearized as previously, and one may have to resort to unary or binary expansions of the flow variables x_a.

The fifth variant arises when the **design of the network** must be decided together with the tolls. In this case, new binary variables y_a stating whether an arc a is installed or not are introduced, and the upper level problem becomes

$$\max_{y,t} \quad \sum_{k \in \mathscr{K}} \eta^k \sum_{a \in \mathscr{A}_1} t_a x_a^k - \sum_{a \in \mathscr{A}_1} f_a y_a$$

where f_a and c_a^k represent the fixed cost for opening arc a and the operating cost for routing commodity k on arc a, respectively. For the sake of consistency, the

constraints

$$x_a^k \leq y_a \quad a \in \mathscr{A}_1, k \in \mathscr{K} \tag{24}$$

must be appended to (20) and (21) in the definition of the sets X^k.

This joint design and pricing problem presents an unusual property for a bilevel optimization problem : constraints (24) can be moved to the first level. Intuitively, this is due to the fact that it is in the leader's interest to adjust its price levels such that the capacity constraints are never active. In technical terms, the dual variable of a capacity constraint will always be null, even if the constraint is tight. This property of the program allows to severely reduce the number of dual variables in the formulation, and preserves the shortest path structure of the lower level problem.

6 Bilevel network interdiction

Interdiction games play an important role in military and drug enforcement applications. In both cases, the goal is to disrupt elements of a transportation network in order to reduce as much as possible the enemy's movements on the network. A network interdiction problem (NIP) involves two actors whose goals are antagonistic. The interdictor or attacker acts first by disrupting some elements of the network. Next, the enemy or defender reacts after having observed the functioning state of the network. In this bilevel setting, the interdictor (leader) anticipates the rational reaction of the defender (follower).

If the game is played only once and the enemy has perfect knowledge of the interdictor's action, both players have no advantage in randomizing their actions (strategies). In other contexts, for instance if the game is repeated or the follower is not perfectly rational, it makes sense to consider mixed strategies, i.e., strategies that assign a probability to each feasible action. This is especially relevant in drug enforcement and terror prevention applications. From now on, we focus on interdiction problems that involve either shortest path or maximum flow lower level problems. The structure of the first variant of path interdiction, which consists in maximizing the shortest path length, makes randomization irrelevant. This is not the case of the second variant, where it may be in the leader's interest to implement mixed strategies.

Throughout, we will make use of the following notation, which is common to the three applications: given a graph $\mathscr{G} = (\mathscr{N}, \mathscr{A})$, we define the set of feasible interdictions as

$$Y = \{y_a \in \{0,1\}, a \in \mathscr{A} : \sum_{a \in \mathscr{A}} f_a y_a \leq B\},$$

where f_a denotes the interdiction cost and B the interdictor's budget.

The shortest path interdiction problem (SPIP)

In SPIP, one associates the traversal cost c_a, as well the additional cost d_a of traversing an interdicted arc. Within its budget limit, the leader selects a subset of interdicted arcs with the aim of maximizing the length of the follower's shortest path from source s to sink t consistent with the leader's design. The formulation of SPIP is:

$$\begin{aligned}
\max_{y\in Y} \quad & \min_{x} \quad \sum_{a\in\mathscr{A}} (c_a + d_a y_a) x_a \\
& \text{subject to} \quad \sum_{a\in\delta(i)^+} x_a - \sum_{a\in\delta(i)^-} x_a = b_i \qquad i \in \mathscr{N} \qquad (25)\\
& \qquad\qquad x_a \geq 0 \qquad\qquad a \in \mathscr{A} \qquad (26)\\
& \qquad\qquad x_a \in \{0,1\} \qquad\qquad a \in \mathscr{A}, \qquad (27)
\end{aligned}$$

where the constant b_i is equal to $+1$ if $n = s$, -1 if $n = t$, and 0 otherwise. This 'max-min' formulation is clearly a particular case of a bilevel program.

The set of feasible solutions of the lower level does not depend on the first level decisions. This can be exploited to develop a single level reformulation based on the value functions (see Section 2) as well as an iterative method to solve SPIP. Let X represent the set of binary vectors corresponding to simple paths from s to t and let $\bar{X} \subset X$. We define a master problem associated to $\bar{X}$ as

$$\begin{aligned}
&\text{MP}(\bar{X}):\\
&\max_{y\in Y} \quad z\\
&\text{subject to} \quad z \leq \sum_{a\in\mathscr{A}} (c_a \bar{x}_a + d_a \bar{x}_a y_a), \quad \bar{x} \in X.
\end{aligned}$$

Problem MP(X) is a valid formulation for SPIP and MP($\bar{X}$) is a relaxation for any $\bar{X} \subset X$ that provides an upper bound on the optimal value of SPIP. On the other hand, given a leader's solution $\bar{x} \in X$, solving the follower's shortest path problem yields a lower bound on the optimal value of SPIP. Further, if this path has a smaller value than the master problem for $\bar{x}$, this new path should be added to $\bar{X}$. Hence, SPIP can be solved by alternating between solving the master and the follower's problem.

Upon replacing the lower level linear program of SPIP by its dual, one obtains the single level formulation

$$\begin{aligned}
\max_{y\in Y,\pi} \quad & \pi_t - \pi_s\\
\text{subject to} \quad & \pi_j - \pi_i - d_a y_a \leq c_a \quad a = (i,j) \in \mathscr{A}.
\end{aligned}$$

It is interesting to note that the iterative procedure described above to solve SPIP amounts to applying Benders decomposition to this last formulation.

A path interdiction problem involving mixed strategies (PIMS)

In this problem, the leader maximizes the probability of catching a follower (evader) whose objective is to travel from a given origin s to a given destination t. The pure strategies of the evader consist in all simple paths from s to t. These can be represented by binary vectors x satisfying (25) and (26). A mixed strategy for the evader is a convex combination of such vectors. Given that the linear system defined by (25) and (26) is totally unimodular , all extreme points of the associated polytope are integer-valued and correspond precisely to these simple paths. In consequence, the set of the evader's mixed strategies is $X = \{x_a, a \in \mathscr{A} : (25) \text{ and } (26)\}$.

The pure strategies of the interdictor consist in subsets of arcs to inspect. If an arc a is inspected by the interdictor and belongs to the path used by the evader, the interdictor catches it with probability p_a. The inspector wants to determine the mixed strategy that maximizes the probability to catch the evader. A mixed strategy for the inspector consists in assigning a probability to each pure strategy.

First, assume that the interdictor can inspect only one arc a at a time, and that p_a represents the probability that it detects the evader if the latter traverses this arc. Its set of mixed strategies is $Y = \{y_a, a \in \mathscr{A} : \sum_{a\in\mathscr{A}} y_a = 1, y_a \geq 0\}$.

The value of the game is given by the probability that the interdictor detects the evader, and PIMS can be formulated as the bilevel program:

$$\min_{x\in X} \max_{y\in Y} \sum_{a\in\mathscr{A}} p_a x_a y_a.$$

According to von Neumann's theorem, an equivalent program is obtained by permuting the min and max operators. The game is actually a zero-sum matrix game. Upon introduction of its value v, it can alternatively be formulated as

$$\begin{aligned} \min_{v,x}\ & v \\ \text{subject to}\quad & (25), (26) \\ & v \geq p_a x_a, \quad a \in \mathscr{A} \end{aligned}$$

which, in our context, posseses an interesting structure.

Consider the maximum value ϕ^* of a flow $\phi^*_a, a \in \mathscr{A}$ in the graph $\mathscr{G}$ with capacities $1/p_a$. By the max flow-min cut theorem of Ford and Fulkerson, $\phi^* = \sum_{a\in\mathscr{C}^*} 1/p_a$, where $\mathscr{C}^*$ is a minimum capacity cut separating s from t. Given that this cut remains minimum if all capacities are multiplied by v, there exists a flow of value 1 in $\mathscr{G}$ as long as v exceeds $1/\phi^*$. Hence the optimal solution of PIMS is $v^* = 1/\phi^*$ and $x^*_a = \phi^*_a/\phi^*$. To retrieve the mixed strategy, i.e., the probability associated to each $s-t$ path, it suffices to use any (polynomial) algorithm that decomposes a flow into a sum of flows on simple $s-t$ paths and cycles. There will be no cycles in our application.

Intuitively, the optimal mixed strategy of the interdictor will only assigns positive probabilities to the arcs of the minimum cut $\mathscr{C}^*$. One can verify that the solution

$y_a^* = 1/(p_a\phi^*)$ for $a \in \mathscr{C}^*$, and $y_a = 0$ otherwise, belongs to Y. Thus, $\sum_{a\in\mathscr{A}} p_a^* x_a y_a^* = \nu^* = 1/\phi^*$, and this solution is optimal.

A natural extension of PIMS involves m independent interdictors. In a pure strategy, several interdictors can be assigned to the same arc. Then, the optimal solution is obtained by multiplying each y_a^* by m, and the evader's solution x^* is unchanged. If, on the contrary, at most one interdictor can be assigned to an arc in a pure strategy, then the set Y must be modified, and the above interpretation in terms of maximum flow does not hold anymore. However, the problem remains polynomial and can be solved as a linear program.

The maximum flow interdiction problem (MFIP)

In MFIP, the leader wants to minimize the maximum flow from a source s to a sink t in a capacitated network. Assuming that arc $\bar{a} = (s,t)$ has an unlimited capacity and that the other arcs $a \neq \bar{a}$ have capacity u_a, the flow from s to t can be seen as a circulation, leading to the following formulation of MFIP:

$$\begin{array}{lll} \min\limits_{y\in Y} & \max\limits_{x} \quad x_{\bar{a}} & \\ & \text{subject to} \quad \sum\limits_{a\in\delta(i)^+} x_a - \sum\limits_{a\in\delta(i)^-} x_a = 0, & i \in \mathscr{N} \\ & \qquad\qquad x_a \leq u_a(1-y_a) & a \in \mathscr{A}\setminus\{\bar{a}\} \\ & \qquad\qquad x_a^k \geq 0 & a \in \mathscr{A} \quad . \end{array}$$

In contrast with SPIP, the feasible solutions of MFIP depend on the upper level variables, hence the Benders decomposition framework does not apply.

Strong duality for the second level problem can again be used to derive a single mixed integer nonlinear formulation:

$$\begin{array}{lll} \min\limits_{y\in Y,\alpha,\beta} & \sum\limits_{a\in\mathscr{A}\setminus\{\bar{a}\}} u_a(1-y_a)\beta_a & \\ \text{subject to} & \alpha_i - \alpha_j + \beta_a \geq 0 & a = (i,j) \in \mathscr{A}\setminus\{\bar{a}\} \qquad (28) \\ & \alpha_t - \alpha_s \geq 1 & \qquad (29) \\ & \alpha_i, \beta_a \in \{0,1\} & i \in \mathscr{N}, a \in \mathscr{A}\setminus\{\bar{a}\} \quad , \qquad (30) \end{array}$$

whose objective is to separate s from t by a cut whose capacity is minimal with respect to the interdicted arcs. Since there exists an optimal solution in which all interdicted arcs belong to the cut, one can rewrite MFIP as

$$\begin{array}{lll} \min\limits_{y\in Y,\alpha,\beta} & \sum\limits_{a\in\mathscr{A}\setminus\{\bar{a}\}} u_a\,\beta_a - \sum\limits_{a\in\mathscr{A}\setminus\{\bar{a}\}} u_a y_a & \\ \text{subject to} & (28)-(30) & \\ & y_a \leq \beta_a & a \in \mathscr{A}\setminus\{\bar{a}\} \quad , \end{array}$$

which is amenable to a branch-and-cut approach. Indeed, the knapsack structure of the set Y, together with the cut configuration can be exploited to derive strong valid inequalities.

7 Bibliographical notes

Bilevel programming is now almost a mature field, with a few monographs and surveys devoted to the area: **?**, **?**, **?**, **?**. Among the various methods that do not guarantee a global solution, we mention the approximation scheme of **?**, as well as the the book edited by **?**, where metaheuristics are addressed.

Introduced by **?**, who proposed for its solution exploratory methods, the continuous version of the network design problem has been further studied by **?**, who analyzed the worst-case behaviour of a class of heuristics. In **?**, the authors proved NP-hardship of the problem, and refined a worst-case bound obtained in **?**. The model that considers improvements to an existing network has been studied by **?**. One of the first mentions of the discrete variant of the problem is due to **?**, who addressed it within a branch-and-bound framework.

Among the limited literature devoted to location within a user-optimized environment, we mention the works of **?**, **?** and **?**, whose models are closely related to the one presented in this chapter. It is also interesting to relate these models to the literature on rational queues, that studies the performance of queueing systems when the agents involve do not (fully) cooperate. A comprehensive survey concerning this topic is available in the monograph by **?**.

The Network Pricing Problem was first considered by **?**, who proposed single level reformulations and and discussed polynomial special cases. **?** and **?** studied the complexity of NPP. **?** proposed some valid inequalities and a branch-and-cut algorithm for NPP. This paper also determines tight values for the big-M constants that enter formulation MILP. **?** presents a branch-and-bound algorithm based on the auxiliary graph introduced by **?**. **?**, relying on a single-level formulation involving both primal and dual variables, proposed a heuristic algorithm that iterates between primal and dual problems.

The description of the clique pricing problem is based on **?** and **?**. The product pricing is discussed in **?** and the parallel with CP is established in **?**. Information concerning the first three variants can be found in **?**, **?** and **?**, and concerning the next two in **?** and **?**. An application of bilevel network pricing worth mentioning involves the management of hazardous material transportation: **?**, **?**.

Interdiction problems date from antiquity: see **?** for a historical perspective. The shortest path interdiction problem has been shown to be NP-hard by **?** and the Benders decomposition method for solving it has been proposed in **?**. The interpretation of the path interdiction problem with mixed strategies is due to **?** who also considered variants of the basic problem. Our presentation of the maximum flow interdiction problem is based on **?**. The problem is shown to be strongly NP-hard and a branch-and-cut algorithm is proposed.